IL/01/105

REC'D 11 APR 2001

WIPO PCT

PA 376375

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

IL 01/105 4

March 14, 2001

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: ***60/245,000***
FILING DATE: ***November 01, 2000***

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

N. WOODSON
Certifying Officer

11-02-00

A/PROV

Please type a plus sign (+) inside this box

Docket Number: 212/01935

PROVISIONAL APPLICATION FOR PATENT COVER SHEET (Small Entity)

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

INVENTOR(S)/APPLICANT(S)		
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Haim Doron	NEERMAN RAJWAN	Hadera, Israel Givataim, Israel

☐ *Additional inventors are being named on page 2 attached hereto*

TITLE OF THE INVENTION (280 characters max)

EFFICIENT RELIABLE UDP USING PACKET DUPLICATOR

CORRESPONDENCE ADDRESS

Direct all correspondence to:

☐ Customer Number [ ] → *Place Customer Number Bar Code Label here*

OR

☒ Firm or Individual Name	William H. Dippert, Esq.				
Address	c/o Cowan, Liebowitz and Latman P.C.				
Address	1133 Avenue of the Americas				
City	New York	State	NY	ZIP	10036-6799
Country	U.S.A.	Telephone	(212) 790-9200	Fax	(212) 575-0671

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒	Specification	*Number of Pages*	7	☐	Small Entity Statement
☐	Drawing(s)	*Number of Sheets*		☐	Other (specify)

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

		FILING FEE AMOUNT
☐	A check or money order is enclosed to cover the filing fees	
☒	The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 03-3419	$75.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____

Respectfully submitted,

SIGNATURE Paul Fenster Date 11/01/2000

TYPED or PRINTED NAME Paul FENSTER REGISTRATION NO. *(if appropriate)* 33,877

TELEPHONE (212) 790-9200

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, DC 20231

P19SMALL/REV04

EXPRESS MAIL CERTIFICATE 37 C.F.R. 1.10

Date of Deposit November 1, 2000

Express Mail Label No. EL 432043944 US

I hereby certify that this paper is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37C.F.R. 1.10 on the date indicated above and is addressed to Commissioner of Patents and Trademarks, Washington, D.C. 20231

William H. Dippert	William H. Dippert
Name of Person Mailing	Signature

Efficient Reliable UDP using Packet Duplicator

Preliminaries

In order to support a large number of broadband movie streams simultaneously, when each stream is, say, 1mbps, full screen, full length, MPEG-2 / MPEG-4 movie, there is a need for a large "farm" of strong servers.

A multi-cast based data distribution system (hereafter "bandwiz system") is described, for example, in a US provisional patent application serial number 60/176,926, a US provisional patent application having serial number 60/217,139, an Israel application having serial number 138114 and an Israel application having a serial number 137624, the disclosures of which are incorporated here-in by reference.
An exemplary described system uses a FEC erasure code to multicast information to clients. The content of the multicast packets, is determined, for example, based on statistics of requests made by the clients, from web servers, from which the data is retrieved for multicasting.

One solution to the problem above is to use Bandwiz system. This system can support streaming to so many users using sometimes a single server. The system requires in a typical example:

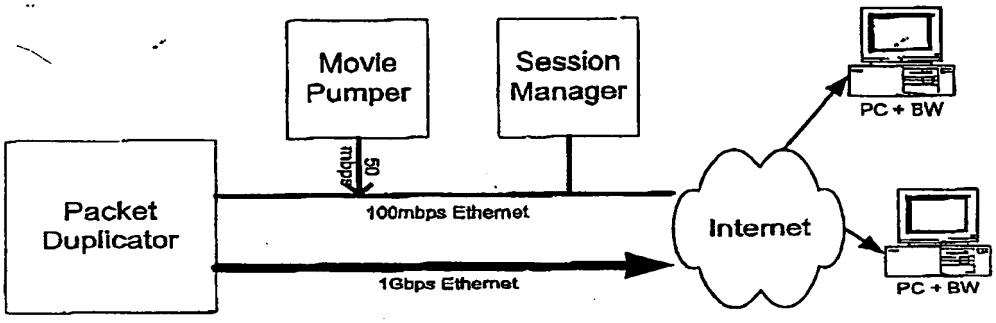
1. Client at the desktop.
2. Multicast IP network between the server and clients.
3. Overlap between the movies.
4. Short delay at the start.
5. Somewhat higher bandwidth at the recciver side.

This disclosure propose a way to eliminate the multicast requirement, by using a packet duplicator.

Application-Level Multicast

The basic idea is to implement application-level multicast, by duplicating packets, using, as one example, the system below.

This system is better than standard video server, because it is easier to duplicate packets than to create them.



Movie Pumper component

This component is part of the bandwiz system mentioned above, where for example, a single PC is capable of pumping 10 different movies, each with expansion factor N=5, with source bit-rate of 1mbps per movie, giving total of 50mbps. This is achievable with reasonable amount of work, or, by using a PC with enough memory to hold all movies simultaneously, i.e., 10GB of memory.

All the packets are standard unicast UDP packets, sent to the packet duplicator. They can be sent, for example, over the 100mbps Ethernet card. A part of the payload of the packet is the "multicast address" of it.

A slightly different approach will be to use an un-modified movie pumper, which sends multicast packets, and use the packet duplicator in order to join that multicast group, and convert it to unicast. In this case, a partially deployed IP multicast can be utilized.

Session Manager component

For each "multicast address" the session manager keeps track of the IP addresses and ports of the connected clients. Each client can send "join" / "leave" / "keep-alive" UDP packet to the session manager, giving the multicast address, the destination IP address and port.

In order to support congestion control for standard FEC, one can add the concept of join to a specific rate. The client will have the ability to join "multicast address", but to get only a given percent of the packets. This can be implemented using the well-known bit reversal mechanism, assuming the manager knows the bitrate of each stream.

Each connection is defined by:

1. The IP address of the manager (possibly pumper).
2. The "multicast address" inside it (say, source UDP port).
3. The IP address of the client.
4. The UDP port in the client.

For each connection, the client can issue a single command, which specifies the connection, the maximal bitrate, and the duration. If the maximal bitrate is zero, it is a leave command. Otherwise, this is a join / rate-change / keep-alive command.

After the duration has passed, the connection will be dropped. This allows us to disconnect clients that were disconnected suddenly.

Packet Duplicator component

The packet duplicator gets the table as defined above, and duplicates incoming packets, from, say, the 100mbps Ethernet, to the 1Gbps Ethernet. We need two interfaces in order to have high utilization of the 1Gbps Ethernet, by eliminating congestion.

The duplicator should be able to achieve the entire 1Gbps capacity of the channel. This can be done, for example, using direct manipulation of drivers in Windows 2000, or by using Free BSD with Zero Copy Sockets implementation..

Please notice that 1GHz PC has more than 4000 clock cycles per packet, which seems more than enough in order to:

1. Get the next IP address, UDP port and bitrate from the table,
2. Check the threshold for rate,
3. Put them in the packet header,
4. Update the checksum,
5. Send the packet using DMA.

Client(s) component

Each client may be a PC with Bandwiz client installed.
One possibility is that the client implement some form of MRCC. The possible modifications from MRCC are:

1. Send "join" / "leave" to the manager, not IGMP commands.
2. Do not implement layers in MRCC. Instead, use the congestion control offered by the packet duplicator.
3. A lot of small changes and simplifications.

Security

We may add some form of security and system stability. For example, in order not to "push" too much data to a client, we may check that each client is not connected for too many streams. For example:

1. Give "busy tone" if there are more than some number of connections.
2. Limit the number of connections a single client can use.
3. Limit the rate that the client can join to.
4. Ask the client for a proof that it has the right rate, by giving some cookies from the previous packets.

Packet Duplicator for "HTTP over Multicast"

Bandwiz system can also designed for general efficient content delivery, i.e., "HTTP over Multicast". The purpose of this system is to accelerate web-site, reduce the output bandwidth, and reduce the number of servers needed to support the request for content from this site. Using the packet duplicator this system can operate without the requirement for native multicast. In this case, the packet duplicator is interfaced with Bandwiz server, where a multicast transmission is translated into packet duplication of coded packets by the packet duplicator. The bandwiz server works, as usual, by adaptively deciding what content to multicast, and how to group content together. Now, when the server decides to multicast a content, it encodes it and start sending it to the packet duplicator which efficiently duplicate the packet to all users who request (in non-overlapping times) this content. Note that the system does not save the output bandwidth, but save server resources, and thus, in general, accelerate the site.

Specification

Finally, below is a description of a possible implementation, which is a combination of the Session Manager and the Packet Duplicator, as defined above. The preferred option is written in *italics*.

Options for packaging:

1. 1U, 19" machine.
2. *Modified Linux kernel, which can run other services as well. The management is in the application level.*
3. Modified Windows 2000 Giga Ethernet drivers, as above.
4. Slow, application level program.

Options for input packets:

1. *Receives unicast UDP packets, in a specific format, containing stream id, packet rate, and payload.*
2. *Receives multicast UDP packets, using IGMP join/leave commands. In this case, the stream id is the multicast group number, and the packet rate is not supported.*

Options for configuration:

1. *The user specifically asks for the stream, by sending UDP "join" commands. This box is responsible of joining real multicast groups, management, security, and limitations. The "join" / "leave" commands should be able to push data to the log file.*
2. There is some configuration machine, instructing the box to send to a given set of users using telnet / SNMP. This box is stupid.

Other (relevant for some machines):

1. There will be an option to connect using telnet / SNMP, and get current state / log files / statistics.
2. Security issues & resource management should be defined.

Claims:

1. A packet duplicator that emulate the operation of a multicast router, but send the packets in unicast to each end user that request the information.
2. A packet duplicator, as in 1, where the packets are coded by FEC.
3. A system for efficient streaming of (near) data on demand, based on coding, multicast and efficient delay, where the packet duplicator implements the multicast.
4. A system for efficient content delivery, based on utilizing overlap in requests to a web site, where the multicasted data is duplicated by the efficient packet duplicator.